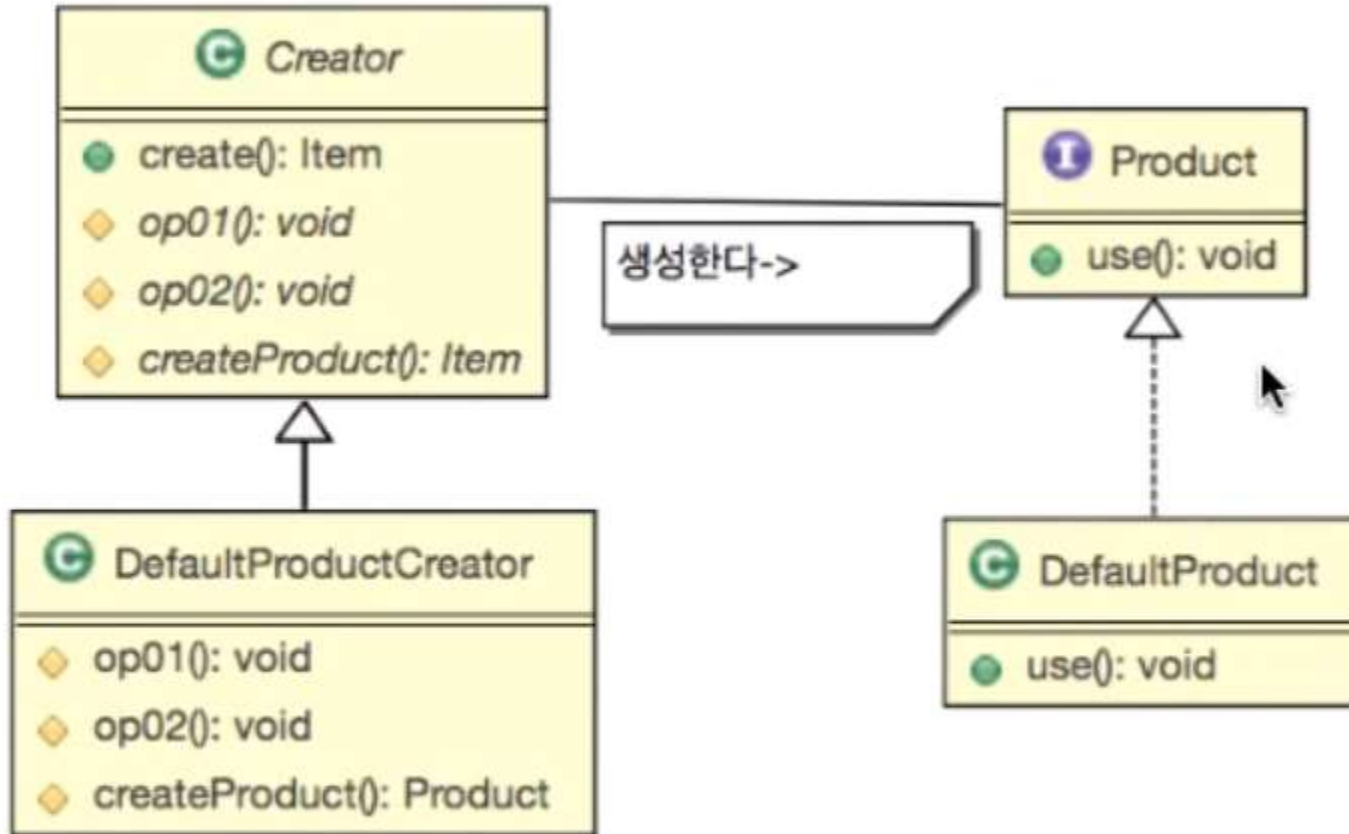


Factory Method Pattern

2019.02.23.

UML

We can know how the template method is used in the factory method.



Interface

```
1  package framework;  
2  
3  public interface Item{  
4      public void use();  
5  }  
6
```

```
1 package framework;
2 public abstract class ItemCreator{
3
4     public Item create(){
5         Item item;
6
7         //step1
8         requestItemsInfo();
9         //step2
10        item = createItem();
11        //step3
12        createItemLog();
13
14        return item;
15    }
16    //Request item information from the database before creating the item.
17    abstract protected void requestItemsInfo();
18    //After the item is created, it is recorded in the database to prevent illegal copying of the item.
19    abstract protected void createItemLog();
20    //It is an algorithm to generate an item.
21    abstract protected void createItem();
22
23 }
```

HpPotion

```
1 package concrete;
2
3 import framework.ItemCreator;
4 public class HpCreator extends ItemCreator{
5     @Override
6     protected void requestItemsInfo(){
7         System.out.println("Retrieve information from the database of health recovery potions.");
8     }
9     @Override
10    protected void createItemsInfo(){
11        System.out.println("Created a new health recovery potion." + new Date());
12    }
13    @Override
14    protected void createItem(){ //Parameter required.
15        //Work
16        return new HpPotion();
17    }
18 }
```

```
1 package concrete;
2
3 public class HpPotion implements Item{
4     @Override
5     public void use(){
6         System.out.println("Magic Recovery!");
7     }
8 }
9 }
```

MpPotion

```
1 package concrete;
2
3 import framework.ItemCreator;
4 public class HpCreator extends ItemCreator{
5     @Override
6     protected void requestItemsInfo(){
7         System.out.println("Retrieve information from the database of magic recovery potions.");
8     }
9     @Override
10    protected void createItemsInfo(){
11        System.out.println("Created a new magic potion." + new Date());
12    }
13    @Override
14    protected void createItem(){ //Parameter필요
15        return new HpPotion();
16    }
17 }
```

```
1 package concrete;
2
3 public class MpPotion implements Item{
4     @Override
5     public void use(){
6         System.out.println("마력 회복!");
7     }
8 }
9
10
```


Main.java

```
1  package concrete;
2  public class Main{
3      public static void main(String[] args){
4          ItemCreator creator;
5          Item item;
6
7          creator = new HpCreator();
8          item = creator.create();
9
10         item.use();
11
12         creator = new MpCreator();
13         creator.create();
14
15         item.use();
16     }
17 }
18
```